Assignment – Microservice Modeling (Version A)

Your company has been contracted to develop software for the management of city trails. In general, a city trail is an urban hiking trail that connects different points of interest. In a meeting, two central business capabilities were identified as microservices: on the one hand, a service for managing the city trails and, on the other hand, a service for managing the individual points of interest (PoI).

In your role of Lead Software Architect of the team, it is your task to create an architecture model for the next meeting.

Structure

During the meeting you took the following notes regarding the data structure of the PoI Service (*if necessary, make sure to model a unique identifier for your data objects*):

- Within the service there is a list of all PoIs.
- A PoI has a name, short description and geolocation.
- A geolocation is a combination of latitude and longitude.

Your notes for the data structure of the Trail Service look like this
(*if necessary, make sure to model a unique identifier for your data objects*):

- The service has a list of trails.
- The service has a list of all PoIs.
- A trail has a name, an expected duration, a description and a list of stops.
- A station consists of a PoI and a sequence number.
- In this context, a PoI has only a name and can be clearly assigned to a PoI from the PoI service.

Interfaces

You have decided on a RESTful HTTP approach for the required interfaces of the two services.
You have noted for the interface of the PoI Service:

- PoIs can be created, read, updated and deleted (*CRUD*).
- PoIs can be searched by entering geocoordinates.

You have noted for the interface of the Trail Service:

- Trails can be created, read, updated and deleted (CRUD).
- A PoI can be added to a trail.
- The list of PoIs can be updated by accessing the PoI Service.

Operation

For the implementation of a first prototype, you have initially chosen Java for both services and would like to loosely couple the services using a service discovery based on the eureka framework. The delivery is to take place per service in the sense of containerization using dockers. The REST interfaces should be accessible via the base URL "https://www.cityhiking.com". Make sure that you specify meaningful access paths (*URI*) for the different services and/or interfaces.

Task 1: UML

Based on the previous case description, model the three views of a microservice architecture using the UML tool Visual Paradigm:

a) Model the domain objects in the form of a UML class diagram.
b) Model the microservices including their interfaces and dependency relationships in the form of a UML component diagram.
c) Model the deployment of the microservices using a UML deployment diagram.

Please commit all modeling artifacts using the folder “Version A” in the e-learning platform.

Task 2: LEMMA

Based on the previous case description, model the three views of a microservice architecture using the set of LEMMA modeling languages:

a) Model the domain objects using the LEMMA Data Modeling Language.
b) Model the microservices including their interfaces and dependency relationships using the LEMMA Service Modeling Language.
c) Model the deployment of the microservices using the LEMMA Operation Modeling Language.

Please commit your exported eclipse project (File → Export → Archive File → Your Project) using the folder "Version A" in the e-learning platform.

//In version B the order of the tasks was switched//